AI Planning
Exercise Sheet 4

Date: 20.11.2014
Students: Axel Perschmann, Tarek Saier

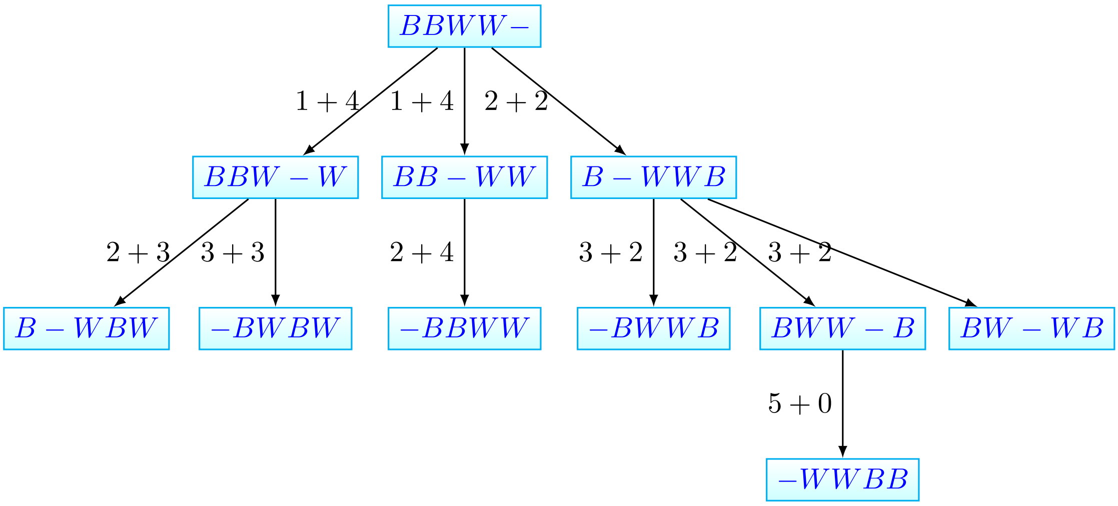
Exercise 4.1

For easy readability let the tiles be referred to as b_1, b_2, w_1 and w_2 and the empty cell be referred to as e. Furthermore, let the actions move and jump be denoted as $m_c(t)$ and $j_c(t)$ respectively where c is the destination cell$\in \{1,2,3,4,5\}$ and t is the tile that is being relocated.
As an example, the initial state is:
b_1, b_2, w_1, w_2, e
If we then apply $j_5(b_2)$ we reach:
b_1, e, w_1, w_2, b_2

(a) Let $\lceil o \rceil$ denote the state s reached by applying the operation $o \in \{m_c(t), j_c(t)\}$.
$f(\lceil m_5(w_2) \rceil) = 1 + 4 = 5$
$f(\lceil j_5(w_1) \rceil) = 1 + 4 = 5$
$f(\lceil j_5(b_2) \rceil) = 2 + 2 = 4$
Apply $j_5(b_2)$ which results in s_1:
b_1, e, w_1, w_2, b_2
$f(\lceil m_2(b_1) \rceil) = 3 + 2 = 5$
$f(\lceil m_2(w_1) \rceil) = 3 + 2 = 5$
$f(\lceil j_2(w_2) \rceil) = 3 + 2 = 5$
$\lceil j_2(b_2) \rceil = I \in closed$
Apply $m_2(b_1)$ which results in s_2:
e, b_1, w_1, w_2, b_2
Apply $m_2(w_1)$ which results in s_3:
b_1, w_1, e, w_2, b_2
Apply $j_2(w_2)$ which results in s_4:
b_1, w_2, w_1, e, b_2
Expanding on s_2:
$\lceil m_1(b_1) \rceil = s_1 \in closed$
$f(\lceil j_1(w_1) \rceil) = 4 + 1 = 5$
$f(\lceil j_1(w_2) \rceil) = 5 + 1 = 6$
Expanding on s_3:
$f(\lceil j_3(b_1) \rceil) = 4 + 1 = 5$
$f(\lceil m_3(w_1) \rceil) = 4 + 2 = 6$
$f(\lceil m_3(w_2) \rceil) = 4 + 2 = 6$
$f(\lceil j_3(b_2) \rceil) = 4 + 3 = 7$

Expanding on s_4:
$f([j_4(b_1)]) = 5 + 0 = 5$

Since h is goal aware and the minimum cost of an operator is 1 we're done at this point. There may be other solutions but none with a cost of less than 5. The resulting plan is: $j_5(b_2), j_2(w_2), j_4(b_1)$ with a total cost of 5 and a final state:
e, w_2, w_1, b_1, b_2



(b) Suppose we loosened the rules of our puzzle in the following way: there are infinitely many empty cells left of the leftmost b_i and right of the rightmost w_j. Furthermore jumps over more than 2 tiles are allowed while the calculation of cost follows the original rule but never exceeds 2.
In this relaxed setting the puzzle can always be solved at a cost of

$$h_r^*(s) = \sum_{i=1}^{\text{number of black tiles}} b_i \times \text{number of white tiles right of } b_i$$

Since this is a less restrictive setting $h_r^*(s) \leq h^*(s)$. Furthermore $h_r^*(s)$ is a more general variant of $h(s)$. In other words, for the specific case I we can say $h_r^*(s) = h(s)$ and therefore $h(s) \leq h^*(s)$.

Exercise 4.2

Start conditions: $\sigma_0 = \{H_x = 1, H_y = 2, G_x = 4, G_y = 4\}$, $h(\sigma_0) = 5$
Possible actions:
$northH, south_H, east_H, west_H, north_G, south_G, east_G, west_G$

1. $improve(\sigma_0)$ iterates over possible successor states of σ_0, until $h(\sigma_1) < h(\sigma_0)$
   $\sigma_1 = \langle \sigma_0, north_H, \{H_x = 1, H_y = 3, G_x = 4, G_y = 4\}\rangle \quad h(\sigma_1) = 4$
   $h(\sigma_1) < h(\sigma_0)$ is causing the end of $improve(\sigma_0)$

2. σ_1 is not a goal-state yet, thus improve(σ_1) is called.
   $\sigma_2 = \langle \sigma_1, east_H, \{H_x = 2, H_y = 3, G_x = 4, G_y = 4\}\rangle \quad h(\sigma_2) = 3$
   $h(\sigma_2) < h(\sigma_1)$ is causing the end of *improve*(σ_1)

3. σ_2 is not a goal-state yet, thus improve(σ_2) is called.
   $\sigma_3 = \langle \sigma_2, east_H, \{H_x = 3, H_y = 3, G_x = 4, G_y = 4\}\rangle \quad h(\sigma_3) = 2$
   $h(\sigma_3) < h(\sigma_2)$ is causing the end of *improve*(σ_2)

4. σ_3 is not a goal-state yet, thus improve(σ_3) is called.
   $\sigma_4 = \langle \sigma_3, south_H, \{H_x = 3, H_y = 2, G_x = 4, G_y = 4\}\rangle \quad h(\sigma_4) = 3$
   $\sigma_5 = \langle \sigma_3, north_G, \{H_x = 3, H_y = 3, G_x = 4, G_y = 5\}\rangle \quad h(\sigma_5) = 3$
   $\sigma_6 = \langle \sigma_3, east_G, \{H_x = 3, H_y = 3, G_x = 5, G_y = 4\}\rangle \quad h(\sigma_6) = 3$

5. No improvement happened, *improve*(σ_3) still running.
   Enforced hill-climbing algorithm will collect all successor states of σ_4, σ_5 and σ_6 for an improvement.
   To make sure this solution sheet will not explode in size, we'll only have a look on the most suitable steps, in this case σ_6
   $\sigma_7 = \langle \sigma_6, south_H, \{H_x = 3, H_y = 2, G_x = 5, G_y = 4\}\rangle \quad h(\sigma_7) = 4$
   $\sigma_8 = \langle \sigma_6, north_G, \{H_x = 3, H_y = 3, G_x = 5, G_y = 5\}\rangle \quad h(\sigma_8) = 4$
   $\sigma_9 = \langle \sigma_6, south_G, \{H_x = 3, H_y = 3, G_x = 5, G_y = 3\}\rangle \quad h(\sigma_9) = 2$
   $h(\sigma_9) < h(\sigma_6)$ is causing the end of *improve*(σ_6)

6. σ_9 is not a goal-state yet, thus improve(σ_9) is called.
   $\sigma_{10} = \langle \sigma_9, south_G, \{H_x = 3, H_y = 3, G_x = 5, G_y = 2\}\rangle \quad h(\sigma_{10}) = 3$

7. No improvement happened, *improve*(σ_9) still running.
   $\sigma_{11} = \langle \sigma_{10}, west_G, \{H_x = 3, H_y = 3, G_x = 4, G_y = 2\}\rangle \quad h(\sigma_{11}) = 2$

8. No improvement happened, *improve*(σ_9) still running.
   $\sigma_{12} = \langle \sigma_{11}, south_G, \{H_x = 3, H_y = 3, G_x = 4, G_y = 1\}\rangle \quad h(\sigma_{12}) = 3$
   $\sigma_{13} = \langle \sigma_{11}, west_G, \{H_x = 3, H_y = 3, G_x = 3, G_y = 2\}\rangle \quad h(\sigma_{13}) = 1$
   $h(\sigma_{13}) < h(\sigma_9)$ is causing the end of *improve*(σ_9)

9. σ_{13} is not a goal-state yet, thus improve(σ_{13}) is called.
   $\sigma_{14} = \langle \sigma_{13}, north_G, \{H_x = 3, H_y = 3, G_x = 3, G_y = 3\}\rangle \quad h(\sigma_{14}) = 0$
   $h(\sigma_{14}) < h(\sigma_{13})$ is causing the end of *improve*(σ_{13})

10. σ_{14} is a goal-state, thus *extract_solution*(σ_{14}) is called.

The solution found by the enforced hill-climbing algorithm:
$north_H, east_H, east_H, east_G, south_G, south_G, west_G, west_G, north_G$
cost/number of steps: 9